

Arquitectura de Control Cognitivo Artificial usando una plataforma computacional de bajo coste.

G. Beruvídes^{a,*}, C. Juanes^a, R. E. Haber^a, F. Castaño^a

^a Centro de Automática y Robótica, Carretera de Campo Real, km 0.200, 28500, La Poveda, Madrid, España.

Resumen

Hoy en día, las principales líneas de investigación tanto en Europa como de EEUU a nivel industrial, abordan aspectos como la interacción hombre-robot y dotar de inteligencia a las máquinas, y por tanto tienen un papel fundamental a la hora de desarrollar cualquier propuesta. Una manera de dotar a las máquinas de conocimiento de la operación que realizan y su interacción con el resto del flujo productivo es la utilización de arquitecturas de control inteligente artificial. A pesar que dichas arquitecturas están dentro de las áreas de investigación priorizadas, aún existen muchas restricciones para su aplicación en la industria de manera general. En este trabajo se propone la emulación de las experiencias socio-cognitivas del ser humano para la toma de decisiones a escala industrial. Las técnicas basadas en Lógica Borrosa, la optimización heurística y las técnicas de auto-aprendizaje desempeñan cada día un papel más importante a la hora de crear los diferentes niveles o capas dentro del sistema. En este trabajo se implementa una arquitectura de control cognitivo artificial enfocada en cuatro aspectos fundamentales: capacidades de auto-aprendizaje y auto-optimización para la estimación; portabilidad y escalabilidad basada en plataformas computacionales de bajo coste; conectividad basada en middleware y enfoque basado en modelos para la estimación y predicción de estados. Finalmente se muestran algunos ensayos de validación en un proceso de microtaladrado que muestran una buena respuesta transitoria y un error de estado estacionario aceptable. Sin lugar a dudas, con la arquitectura de control cognitivo artificial propuesta se sientan las bases para su futura aplicación en una instalación industrial.

Palabras Clave:

Control cognitivo, Auto-aprendizaje, Auto-organización, Computación de bajo coste.

1. Introducción

Existe una abundante literatura relacionada con los sistemas basados en el conocimiento y las arquitecturas cognitivas artificiales (Haber et al. 2002). Sin embargo, la aplicación real de las arquitecturas cognitivas artificiales en la industria está todavía en una fase embrionaria. Una arquitectura cognitiva híbrida que se basa en la integración de enfoques emergentes y cognitivistas utilizando estrategias evolutivas se propone en (Romero Lopez 2010) con un nivel cognitivo inspirado en sistemas inmunológicos artificiales basados en algoritmos genéticos. Bannat et al. (2011) presentaron un artículo seminal sobre la forma artificial la cognición se puede aplicar en los sistemas de producción. Los autores observaron que los sistemas de control con autoaprendizaje y auto-optimización son claves para los sistemas cognitivos artificiales. Sanchez-Boza et al. (2011) proponen una arquitectura de control cognitivo artificial basada en el Modelo de los Circuitos Compartidos (SCM). Su principal inconveniente es la falta de procedimientos sistemáticos para el aprendizaje y la optimización en la arquitectura de cinco capas que proponen.

A pesar de la importancia de las arquitecturas cognitivas como un área de investigación, las estrategias para la aplicación del control cognitivo artificial a nivel industrial tienen muchas limitaciones y hay muy pocos trabajos formales desde la perspectiva de la ingeniería de control (Kopácsi et al. 2013). Por otra parte, los aspectos relevantes de las arquitecturas de

control cognitivo aún no se han abordado suficientemente bien debido a las limitaciones de las aplicaciones en red y a utilización de técnicas de inteligencia artificial no claramente justificadas (Haber-Guerra et al. 2008, Martin and Haber, 2009). En primer lugar, la interacción entre el auto-aprendizaje y la auto-optimización no está abordada de forma explícita en muchas arquitecturas. En segundo lugar, los procedimientos de evaluación de las arquitecturas cognitivas son limitados y su disponibilidad está con frecuencia restringida. Los criterios de evaluación y los índices de rendimiento, no son de fácil aplicación en la industria. Se requiere por tanto asociar la definición de cifras de mérito relacionadas con el comportamiento transitorio, los sistemas de estado dinámico y constante, y el esfuerzo de control, entre otros, todos los cuales dificulta cualquier comparación de las capacidades de hoy en día y el rendimiento de estas arquitecturas (Penedo et al. 2012). Por último, muchas arquitecturas cognitivas carecen de inspiración biológica o las bases desde la perspectiva de las neurociencias son insuficientes. Es esencial que la implementación computacional de arquitecturas tenga raíces biológicas y psicológicas con un enfoque basado en la Teoría de Sistemas y la Automática (Sanchez Boza and Haber Guerra 2011).

A su vez, existen una amplia gama de métodos deterministas y estocásticos para resolver problemas de optimización. Muchos métodos de optimización se pueden aplicar para esta tarea que van desde algoritmos genéticos para

* Autor en correspondencia.

Correo electrónico: gerardo.beruvídes@car.upm-csic.es (G. Beruvídes)

la optimización de partículas enjambre (Zhang et al. 2009, Wang and Yang 2009). Por otra parte, los algoritmos evolutivos (EA) han demostrado su idoneidad como un método para la optimización multi-objetivo. EA mantener una familia de soluciones durante el proceso de optimización, que tienen el potencial de almacenar un conjunto de soluciones de comercio-off simultáneas con el potencial para aprovechar las sinergias de una búsqueda paralela en todas las soluciones posibles. Sin embargo, las AE se evalúan por lo general experimentalmente usando diversos problemas de prueba, porque una evaluación analítica de su comportamiento es muy complicada. Por lo tanto, su rendimiento en problemas aleatorios no se puede garantizar antes de la aplicación (Huband et al. 2006).

Los procesos de fabricación a escala micro es un claro ejemplo de un sistema dinámico que opera en un entorno caracterizado por el cambio continuo, siendo un escenario perfecto para probar la estrategia de control cognitivo artificial propuesta. En este escenario, uno de los principales objetivos es el desarrollo de tecnologías y algoritmos que permitan un sistema de control más rápido con capacidades de auto-optimización y auto-aprendizaje. Estos procesos de fabricación se caracterizan por la presencia de la dinámica no lineal y el tiempo-variantes que surgen a partir del comportamiento de la temperatura, fuerzas, torque y otras variables representativas; características que aumentan la complejidad funcional de la microfabricación y los requisitos de precisión de los sensores, actuadores y los recursos informáticos (Precup et al. 2015).

En este estudio, se describe la arquitectura de control cognitivo artificial con capacidades de auto-optimización y auto-aprendizaje, así como su simulación y aplicación en tiempo real para el control de la fuerza de los procesos de microfabricación. La arquitectura, está basada en el modelo de las habilidades socio-cognitivas (Kit et al. 2013) y toma los principios de simplicidad y escalabilidad en cuenta. Otro desafío es implementar la arquitectura en un lenguaje de programación portátil, así como la utilización de una plataforma de bajo coste computacional.

2. Métodos de Auto-aprendizaje y Auto-optimización

En esta sección, se presentan los algoritmos de auto-optimización y auto-aprendizaje desarrollados. El aprendizaje por refuerzo pertenece a una categoría de técnicas de aprendizaje no supervisado (Kaelbling et al. 1996) a diferencia de las técnicas de aprendizaje supervisado (Alique et al. 2000). Es un paradigma de aprendizaje con el aprendizaje por recompensas / sanciones con algunas aplicaciones interesantes para el control de sistemas complejos, a fin de maximizar las medidas de desempeño numéricos que expresan un objetivo a largo plazo (Szepesvári 2010).

Este trabajo se centra en el algoritmo Q-learning. La razón principal detrás de esta elección es la sencillez de su planteamiento, su característica modelo libre y los resultados de este algoritmo descrito en la literatura. Q-learning puede ser usado para encontrar una política de selección de una acción óptima para cualquier dado (finito) el proceso de decisión de Markov. Se lleva a cabo mediante el aprendizaje de una función de acción-valor que finalmente genera la utilidad

esperada de tomar una acción determinada en un estado determinado y luego sigue la política óptima. Además, Q-learning puede manejar problemas con transiciones estocásticas y recompensas, sin mayor adaptación.

El modelo de Markov, consiste en un número de estados S y un conjunto de acciones por estado A . Mediante la realización de cada acción A , el agente puede pasar de un estado a otro. La ejecución de una acción en un estado específico proporciona el agente con una recompensa. Siendo el objetivo del agente maximizar la recompensa total quedándose con el aprendizaje de la mejor acción para cada estado. Por tanto, el algoritmo tiene una función que calcula la calidad de una combinación estado-acción, $Q: S \times A \rightarrow R$

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t)(R_{t+1} + \gamma \max_{a \in A} Q_t(s_{t+1}, a) - Q_t(s_t, a_t)) \quad (1)$$

donde, S_t es el estado en el momento t ; a_t es la acción tomada en el momento t ; R_{t+1} es la recompensa recibida después de realizar la acción a_t ; α_t es la tasa de aprendizaje y γ es el factor de descuento que opera cola comparación de las recompensas actuales y anteriores.

Antes de comenzar el aprendizaje, Q puede devolver cualquier valor fijo, elegido por el diseñador del problema. Entonces, cada vez que el agente selecciona una acción, que recibe su recompensa y entra en el nuevo estado. El núcleo del algoritmo es una actualización del valor de iteración simple. Se asume que el valor antiguo y hace una corrección basada en la nueva información.

En la arquitectura cognitiva como en cualquier enfoque jerárquico hay diferentes escalas de tiempo (anchos de banda). El procedimiento de aprendizaje se ejecuta a una frecuencia más baja que el mecanismo de control, que se asemeja a un concepto de cascada, porque el proceso tiene que funcionar durante un período de tiempo suficiente, para un aprendizaje correcto. Teniendo en cuenta este factor, si el mecanismo de control tiene un tiempo de muestreo para el control de $P_control$, el aprendizaje tiene que ser realizado por lo menos diez veces más lento que el control, es decir, $P_learning = \delta P_control$, donde delta tiempo (δ) $\delta \in \mathbb{N}$, $\delta \geq 10$. La figura 1 muestra todas las modificaciones realizadas al algoritmo.

Algoritmo 1: Modificaciones realizadas al método de Q -learning

```

1 Inicializar  $Q(s_t)$  arbitrariamente o con valores obtenidos por algún
  método anterior
2 Inicializar  $s_0$  con un valor arbitrario
3 repeat
4   foreach step do
5       Seleccionar  $a_t$  usando el algoritmo de  $\epsilon$ -greedy;
6       Realizar acción  $a_t$  y cambiar a  $s_{t+1}$ ;
7       Esperar  $\delta p_{control}$  y recibir  $R$ ;
8       Actualizar  $Q$ -values;
9        $s_t \leftarrow s_{t+1}$ ;
10  end
11 until  $s_t$  sea un estado final;
```

Figura 1: Modificaciones propuestas al algoritmo de Q -learning.

La configuración óptima de estrategias controlador difuso basado en la optimización basada en gradiente estocástico se informó en diferentes obras (Barchinezhad 2014). Sin embargo, muchas de estas técnicas de optimización aún no se han aplicado en los procesos industriales reales, debido a la gran complejidad de los algoritmos de optimización, la necesidad de definir las funciones apropiadas de costes e índices de rendimiento, el comportamiento apropiado y / o la falta de fórmulas empíricas para su uso en industria. Desde un punto de vista coste computacional, el importe de la heurística en comparación con otros métodos de optimización de gradiente libre se reduce, lo que significa que, en la práctica, no se utilizan muchos métodos de optimización heurística (Chen et al. 2010).

Por todas las razones anteriores, hemos seleccionado el método denominado entropía cruzada (*Cross Entropy*, CE) (Rubinstein 2008, 2005), como el algoritmo principal la optimización de la arquitectura de control cognitivo artificial. El cálculo analítico de las reglas de actualización de densidades es una de las características más atractiva de la entropía cruzada, haciéndolos extremadamente eficiente y rápido. Además, el marco teórico a CE permite a los estudios teóricos del método, que puede proporcionar directrices de sonido sobre la aplicabilidad de este algoritmo en arquitecturas cognitivas artificiales. La figura 2 muestra el algoritmo propuesto.

Algoritmo 2: Algoritmo de Entropía Cruzada

```

1 Inicializar  $v_0$ .
2 Generar muestras  $x_t$  de  $i = 1 \dots n$ , para una función de densidad  $g(x_t, v_t)$ .
3 Actualizar  $v_t$ .
4 Repetir paso 2 hasta que converga el algoritmo o terminen
   $n$ -iteraciones definidas inicialmente.
5 Si la convergencia se ha logrado  $t = t^*$ , teniendo la variable aleatoria
  seleccionada para  $g(x_t, v_t)$  toda su masa concentrada en  $x^*$ .
```

Figura 2: Algoritmo de entropía cruzada propuesto.

Donde, x es una variable aleatoria, f es una función definida $f(x \rightarrow \mathbb{R})$ y v es la simulación de eventos no extraños alrededor del óptimo. Finalmente, los problemas de optimización desde un punto de vista de ingeniería, tendrán restricciones dependiendo del proceso, por lo tanto los límites imponentes sobre la función (f) de distribución para la generación de muestras, deben asegurar que el muestreo esté dentro de la región apropiada.

3. Arquitectura de control cognitivo propuesta

La arquitectura cognitiva artificial propuesta debe cumplir con dos requisitos fundamentales: los funcional (RF) y los no funcionales (RNF). Los principales requisitos funcionales pueden resumirse como:

RF1 Arquitectura de control: la función principal de esta arquitectura es el control de procesos; la implementación de la arquitectura debe permitir al usuario asignar un proceso y que esta se prepare para controlarlo.

RF2 Modelos: la arquitectura tiene varios modelos que sirven para controlar un proceso con diferentes procedimientos. Hay cuatro tipos de modelos: los modelos de lazo simple, modelos directos, modelos inversos y modelos de simulación. La configuración de los modelos directos e inversos se inspira en el paradigma de control modelo interno muy conocido en la comunidad de Ingeniería de Control, pero con bases científicas en la interacción cerebro-cerebelo (Ito 2008).

RF3 Modos: la arquitectura debe funcionar en diferentes modos. Un modo está definido por una configuración predefinida de los diferentes elementos de la arquitectura (modelos, valores de referencia y entidad de proceso) para controlar un proceso. Un mecanismo permite a la aplicación para cambiar entre los modos mientras se ejecuta. La conmutación puede ser suavizada por un filtro de primer orden para garantizar una transición sin problemas de un modo a otro.

RF4 Adaptación: la aplicación debe proporcionar un componente para la selección de los modelos requeridos por un modo específico. La elección del componente puede ser para lograr diferentes objetivos.

RF5 Optimización: la arquitectura debe proporcionar la funcionalidad para el ajuste óptimo de los modelos de control sobre la base de un modelo de simulación del proceso físico. Con esta acción, la arquitectura será capaz de mejorar su comportamiento durante la ejecución de los diferentes procesos.

RF6 Aprendizaje en línea: la arquitectura debe proporcionar un mecanismo para ejecutar un algoritmo de aprendizaje. Una vez más, este mecanismo mejorará el comportamiento del sistema en general.

RF7 Objetivos: la arquitectura debe garantizar que el usuario introduce los objetivos a alcanzar, por ejemplo, la productividad, el rendimiento, etc.

RF8 Tipos de datos: la arquitectura debe permitir diferentes tipos de datos, como el número entero, doble o de cadena.

Los principales requisitos no funcionales de la arquitectura se describen a continuación:

RNF1 Middleware: la arquitectura deberá ser genérica y flexible para permitir al usuario utilizarla sobre un middleware, por ejemplo, el usuario puede desear utilizar la arquitectura para controlar un proceso en un lugar diferente, es decir, para distribuir la arquitectura de control en proceso remoto.

RNF2 Extensibilidad: la arquitectura se ha diseñado para facilitar las tareas de la adición de los modelos, algoritmos de control, mecanismos de optimización y de aprendizaje, etc.

La figura 3a muestra un esquema general de arquitectura propuesta. Está dividida en dos niveles: cognitivo (ver figura 3b) y ejecutivo (Ver figura 3c). Para obtener una mejor comprensión de la interconexión entre las diferentes partes de la arquitectura en la figura 3b se puede apreciar que hay un espejo del nivel de ejecución dentro del nivel cognitivo, siendo necesario para los mecanismos de organización, de adaptación y de aprendizaje que hacen uso de las simulaciones en tiempo real. De lo contrario estos mecanismos tendrían que utilizar el tiempo de procesamiento de la unidad de ejecución, lo que limita el rendimiento en tiempo real.

Por otro lado, en la unidad ejecutiva, la entrada más importante es los cambios de la configuración (ver figura 3a) que sirven para introducir modificaciones tanto en la configuración como en los parámetros. Dichas modificaciones de los parámetros cambian los valores de un modelo en particular (el modelo i) del mecanismo de aprendizaje y los cambios en su configuración pueden ser de dos tipos: nueva asignación de un modelo a un modo específico o cambiar entre los modos.

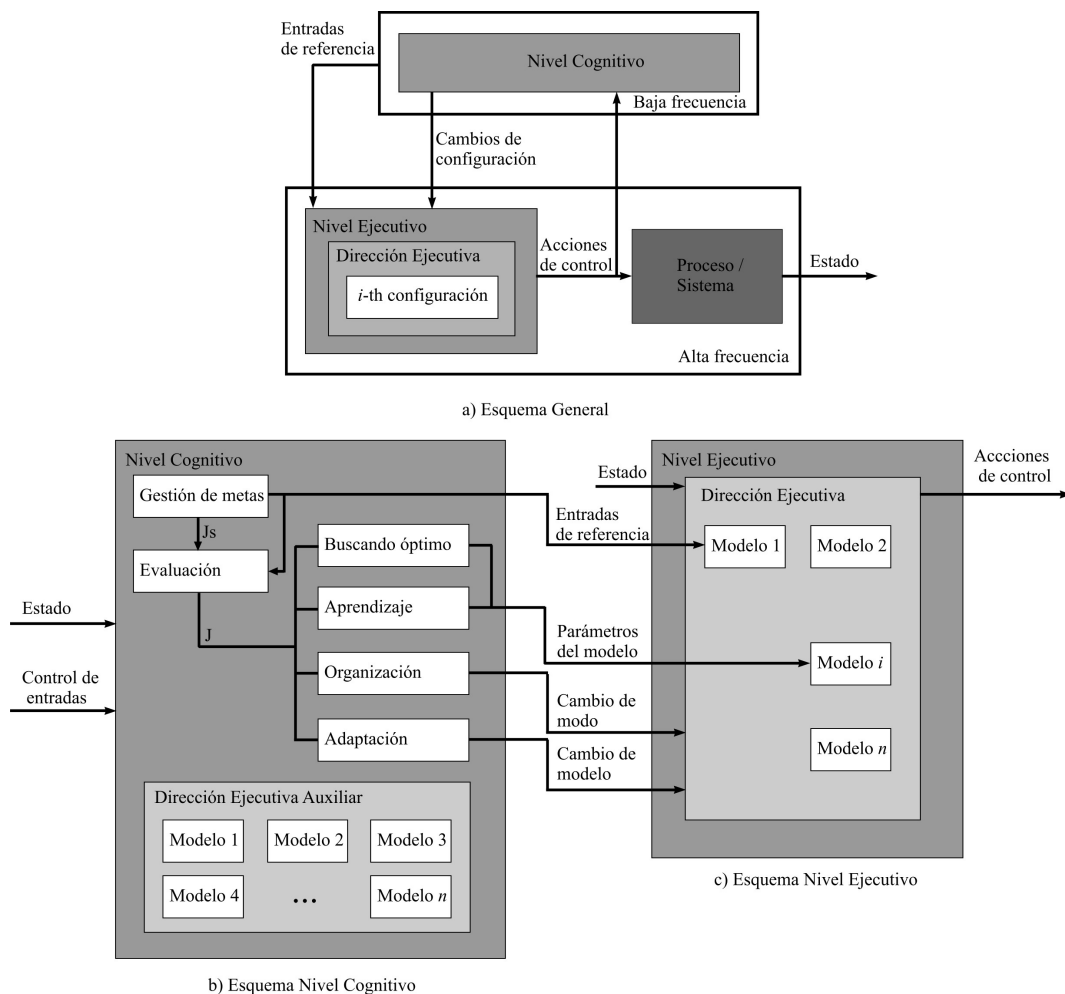


Figura 3: Arquitectura de Control Cognitivo propuesta.

Otro aspecto interesante de la propuesta, es la utilización de plataformas de computación de bajo coste. Para la confección de la arquitectura cognitiva se revisaron tres de las plataformas de computación de bajo coste más populares reportados en la literatura: Raspberry Pi Modelo B, HummingBoard-i2 y BEAGLEBONE Negro.

Raspberry Pi es capaz de realizar todas las tareas de un ordenador de escritorio, desde navegar por Internet, reproducción de videos de alta definición, uso de hojas de cálculo, procesadores de texto hasta juegos (Raspberry Pi model B specifications). Este pequeño dispositivo de bajo coste es muy versátil, está equipado con un procesador de un solo núcleo ARM v6 y 512 MB SDRAM 700 MHz. Raspberry Pi puede interactuar con el medio ambiente y los dispositivos en una amplia gama de proyectos fabricante digitales.

El HummingBoard-i2 y Raspberry Pi ambos comparten un diseño y configuración muy similar. El primero representa una buena opción, si los usuarios están buscando una opción más potente con un 1,0 GHz de doble núcleo ARM v7 y 1 GB SDRAM a más del doble del coste de Raspberry Pi.

BEAGLEBONE Negro es adecuado para usuarios que buscan un poco más de potencia que el Raspberry Pi, el más fácil de configurar, fácil comercialización, o tener una necesidad de interactuar con muchos sensores externos. Su configuración consiste en un AM335x 1GHz ARM Cortex-A8 y 512 MB de memoria RAM DDR3 a un coste similar a Raspberry Pi.

Finalmente se decidió Raspberry Pi Modelo B, en parte porque es la plataforma más popular, con una comunidad de usuarios muy activa, además uno de los objetivos es desarrollar una herramienta capaz de controlar cualquier proceso al menor coste posible, siendo la Raspberry Pi de las 3 la plataforma menos costosa. Aunque no se descarta la utilización de otras plataformas de bajo coste en el futuro.

4. Caso de estudio

Debido a las pequeñas dimensiones de los procesos de microtaladrado, el control en línea es muy difícil de llevar a cabo, dificultades que son también relevantes en el taladrado convencional (Del Toro et al. 2007). Los sistemas de control de calidad en línea pueden proporcionar información en tiempo real, que puede ser suministrado como una retroalimentación a CNC para los parámetros de ajuste de corte. Este tipo de sistemas se ha propuesto para el mecanizado convencional (Beruvides et al. 2013, Beruvides et al. 2014).

Para la validación de la arquitectura propuesta se utilizó como caso de estudio un proceso de microtaladrado. Todas las operaciones de corte se realizaron en un centro de la máquina Ultra-Precisión Kern Evo, equipado con un Heidenhain

iTNC540 CNC, con una velocidad máxima del husillo de 50 000 rev/min y la velocidad de avance de 16 000 mm/min, respectivamente. La medición de las señales de fuerza de corte se realizó con un sensor de Kistler (MiniDyn 9256C1) se utilizó en el eje z, con un ancho de banda de hasta 5 kHz. Todas las señales de los sensores se alimentaron a un 6251 de National Instruments tarjeta de adquisición de datos NI, con una velocidad de muestreo de 50 kHz, y fueron procesadas con controlador embebido PXI-8187 de National Instruments. La posición de la punta de la herramienta (x, y, z) se obtuvo a través de la conexión ethernet de la CNC de la máquina, utilizando una frecuencia de muestreo cada 12 ms (83,33 Hz). La Figura 4 muestra la plataforma con los principales dispositivos que están etiquetados. Raspberry Pi (Pi 1) y Raspberry Pi (Pi 2) ejecutan la parte cognitiva y ejecutiva de la arquitectura, respectivamente. El *Process Host* obtiene las salidas de proceso de la máquina KERN Evo y las envía a la arquitectura a través del Middleware *ZeroC Ice*, luego de recibir la acción del control de la arquitectura lo envía a la máquina KERN Evo. Como se muestra en la imagen, la comunicación entre el cliente y el proceso de KERN Evo se realiza a través de Ethernet. El programa de registro IceGrid está ejecutando de forma permanente en el host del registro para permitir a los diferentes hosts que se identifiquen entre sí. Por último, el cliente puede utilizar la interfaz gráfica de usuario desarrollada para interactuar con los diferentes componentes.

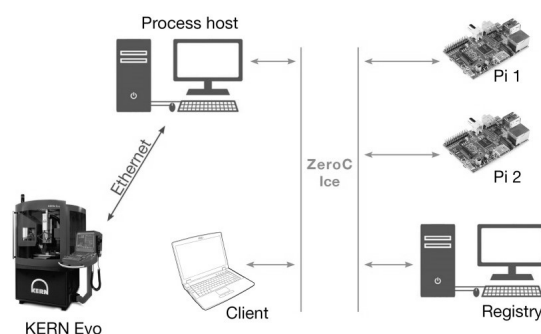


Figure 4: Diagrama de conexión proceso-arquitectura

En el caso de estudio en particular, existen restricciones de tiempo real y retraso en las comunicaciones. El tiempo de corte es de alrededor de 5s por agujero con las condiciones de corte utilizados. Por esta razón se dificulta el control del proceso en tiempo real y existe un pequeño retraso entre el centro de mecanizado y la arquitectura cognitiva.

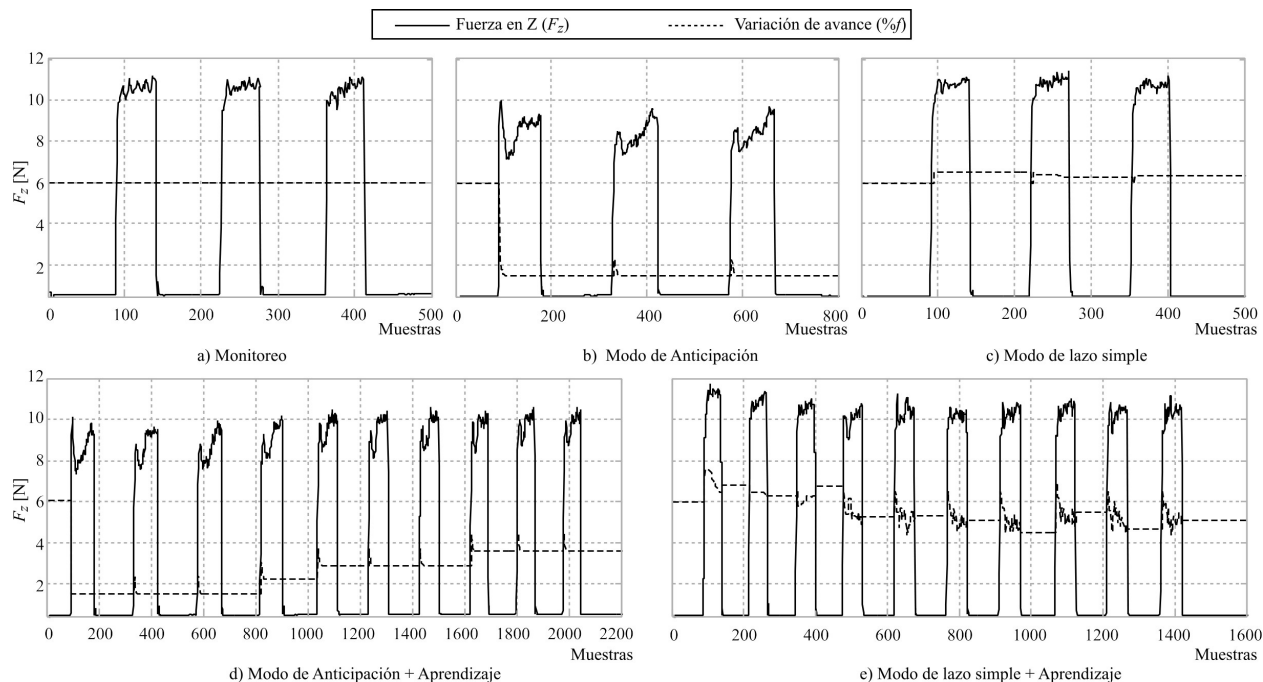


Figure 5: Valores alcanzados para las diferentes estrategias de control

En primer lugar, para supervisar el comportamiento del proceso sin la arquitectura de control cognitivo se realizaron experimentos en tres situaciones: solo Monitorización, Modo de anticipación y Modo de lazo simple. El punto de ajuste preestablecido para el control de la fuerza fue de 10N. Finalmente, se utilizaron conjuntos de diez taladros para validar el algoritmo de aprendizaje.

Los resultados de estos experimentos se representan en la Figura 5. La Figura 5a) muestra el comportamiento de la fuerza sin control, con un aumento evidente más allá de 10N. En un escenario, con el modo de control inverso (anticipación) (ver Figura 5b), la señal de control no puede alcanzar el valor de consigna (10N), mientras que con el modo de lazo simple (ver Figura 5c) el comportamiento de la fuerza supera el punto de referencia incluso más allá del estado transitorio.

Por el contrario, si se activa el algoritmo de aprendizaje modificándose en cada iteración y se mejora el rendimiento general del control cognitivo artificial. Los resultados corroboran cómo el algoritmo Q-learning contribuye a una mejor respuesta.

En la figura 5 se puede observar cómo los modos de anticipación + aprendizaje (ver Figura 5d) y un lazo simple + aprendizaje (ver Figura 5e) mantienen el valor de la fuerza de corte sobre el valor prescrito (10N). Este resultado valida la utilidad del algoritmo de aprendizaje dentro de la arquitectura de control cognitivo y la utilidad de este en los procesos industriales.

5. Conclusiones

En este trabajo se ha presentado el diseño e implementación de un sistema de control cognitivo artificial en una plataforma de computación de bajo coste con capacidades de auto-optimización y auto-aprendizaje. Más allá del caso de estudio de caso sobre el control de la fuerza para los procesos microtaladrado y los resultados presentados, ha demostrado la idoneidad de la aplicación en una instalación industrial.

Plenamente consciente de la naturaleza preliminar de este estudio, los objetivos en un futuro próximo apuntan hacia el desarrollo de nuevos modelos para crear un repositorio capaz de realizar tareas más complejas. Además, se pretende mejorar la forma de ejecutar los componentes para un mejor funcionamiento de esta nueva plataforma de bajo coste.

English Summary

Artificial Cognitive Control Architecture using Low Cost Computing.

Abstract

Nowadays, the main research lines of research in Europe and US at industrial level address issues such as human-robot interaction and provide intelligence to machines, taking a key role in developing any new research proposal. The use of artificial cognitive control system is one way of providing knowledge for the machines and it makes easy its interaction with the rest of machines in the production flow. Although these architectures are within the priority areas of research, there are still many restrictions on its application in industry in general. These are based on the emulation of socio-cognitive processes developed by man for taking decisions on an

industrial scale. The techniques of fuzzy logic, artificial intelligence, optimization *a priori* or *a posteriori*, self-learning techniques, etc., take increasingly large role in creating the different levels or layers within the system. In this work an artificial cognitive monitoring architecture focused on four key aspects is implemented: self-learning and self-optimization for estimation; low cost computing platform; connectivity middleware-based models and models for the estimation and prediction of states. Finally, validation tests were developed in microdrilling process, obtaining a good transient response and an acceptable steady state error. Undoubtedly, with the artificial cognitive control architecture is useful for future application in industrial facilities.

Keywords:

Cognitive control; Self-learning; Self-optimization; Low cost computing.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad a través de su proyecto de investigación DPI2012-35504 CONMICRO.

Referencias

- Alique, A., Haber, R. E., Haber, R. H., Ros, S., & Gonzalez, C. (2000) Neural network-based model for the prediction of cutting force in milling process. A progress study on a real case. Paper presented at the IEEE International Symposium on Intelligent Control - Proceedings, pp. 121-125
- Bannat, A., Bautze, T., Beetz, M., Blume, J., Diepold, K., Ertelt, C., Geiger, F., Gmeiner, T., Gyger, T., Knoll, A., Lau, C., Lenz, C., Ostgathe, M., Reinhart, G., Roesel, W., Ruehr, T., Schuboe, A., Shea, K., Stork Genannt Wersborg, I., Stork, S., Tekouo, W., Wallhoff, F., Wiesbeck, M. and Zaeh, M.F. (2011) Artificial cognition in production systems. IEEE Transactions on Automation Science and Engineering 8(1), 148-174.
- Barchinezhad, S.E., Mahdi (2014) A New Fuzzy and Correlation Based Feature Selection Method for Multiclass Problems. International Journal of Artificial Intelligence 12(2), 24-41.
- Beruvides, G., Quiza, R., Del Toro, R., Haber, R. E. (2013) Sensing systems and signal analysis to monitor tool wear in microdrilling operations on a sintered tungsten-copper composite material. Sensors and Actuators, A: Physical 199, 165-175. doi:10.1016/j.sna.2013.05.02
- Beruvides, G., Quiza, R., del Toro, R., Castaño, F. and Haber, R. (2014) Correlation of the holes quality with the force signals in a microdrilling process of a sintered tungsten-copper alloy. International Journal of Precision Engineering and Manufacturing 15(9), 1801-1808.
- Chen, T., Tang, K., Chen, G. and Yao, X. (2010) Analysis of computational time of simple estimation of distribution algorithms. IEEE Transactions on Evolutionary Computation 14(1), 1-22.
- Del Toro, R. M., Schmittiel, M. C., Haber-Guerra, R. E., & Haber-Haber, R. (2007) System identification of the high performance drilling process for network-based control. Paper presented at the 2007 Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2007 vol. 1 PART B, pp. 827-834.
- Haber, R. E., Haber, R., Alique, A., Ros, S. (2002) Application of knowledge-based systems for supervision and control of machining processes. Handbook of software engineering and knowledge engineering, 2, 673-710.
- Haber, R. E., K. Cantillo, and J. E. Jiménez (2005) Networked sensing for high-speed machining processes based on CORBA. Sensors and Actuators, A: Physical 119(2):418-426.
- Haber-Guerra, R. E., Haber-Haber, R., Andrés, D. M., & Palomar, A. A. (2008) Networked fuzzy control system for a high-performance drilling process. Journal of Manufacturing Science and Engineering, Transactions of the ASME, 130(3), 0310091-0310096. doi:10.1115/1.2783280
- Huband, S., Hingston, P., Barone, L. and While, L. (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10(5), 477-506.
- Kaelbling, L.P., Littman, M.L. and Moore, A.W. (1996) Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237-285.
- Kit, D., Ballard, D.H., Sullivan, B. and Rothkopf, C.A. (2013) A hierarchical modular architecture for embodied cognition. Multisensory Research 26(1-2), 177-204.
- Kopácsi, S., Kovács, G.L. and Nacs, J. (2013) Some aspects of dynamic 3D representation and control of industrial processes via the Internet. Computers in Industry 64(9), 1282-1289.
- Ito, M. (2008) Control of mental activities by internal models in the cerebellum, pp. 304-313.
- Martin, A. G., Guerra, R. E. H. (2009) Internal model control based on a neurofuzzy system for network applications. a case study on the high-performance drilling process. IEEE Transactions on Automation Science and Engineering 6(2), 367-372 doi:10.1109/TASE.2008.2006686
- Penedo, F., Haber, R. E., Gajate, A., Del Toro, R. M. (2012). Hybrid incremental modeling based on least squares and fuzzy K-NN for monitoring tool wear in turning processes. IEEE Transactions on Industrial Informatics, 8(4), 811-818. doi:10.1109/TII.2012.2205699.
- Precup, R.-E., Angelov, P., Costa, B.S.J. and Sayed-Mouchaweh, M. (2015) An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. Computers in Industry (0).
- Raspberry Pi model B specifications <http://docs-europe.electrocomponents.com/webdocs/127d/0900766b8127da4b.pdf>
- Rubinstein, R. (2008) Semi-iterative minimum cross-entropy algorithms for rare-events, counting, combinatorial and integer programming. Methodology and Computing in Applied Probability 10(2), 121-178.
- Rubinstein, R. (2005) A stochastic minimum cross-entropy method for combinatorial optimization and rare-event estimation. Methodology and Computing in Applied Probability 7(1), 5-50.
- Romero Lopez, O.J. (2010) Applications of Evolutionary Computation, Pt I, Proceedings. DiChic, C., Cotta, C., Ebner, M., Ekart, A., EsparciaAlcazar, A.I., Goh, C.K., Merelo, J.J., Neri, F., Preuss, M., Togelius, J. and Yannakakis, G.N. (eds), pp. 392-401.
- Sanchez Boza, A., Haber Guerra, R. and Gajate, A. (2011) Artificial cognitive control system based on the shared circuits model of sociocognitive capacities. A first approach. Engineering Applications of Artificial Intelligence 24(2), 209-219.
- Sanchez Boza, A. and Haber Guerra, R. (2011) Connectionist Models of Behavior and Cognition, World Scientific Publishing.
- Szepesvári, C. (2010) Algorithms for reinforcement learning. Synthesis Lectures on Artificial Intelligence and Machine Learning 4(1), 1-103.
- Wang, Y. and Yang, Y. (2009) Particle swarm optimization with preference order ranking for multi-objective optimization. Information Sciences 179(12), 1944-1959.
- Zhang, J., Zhuang, J., Du, H. and Wang, S. (2009) Self-organizing genetic algorithm based tuning of PID controllers. Information Sciences 179(7), 1007-1018.